

**REMARKS**

Claims 14-33 are pending.

Claims 14, 15-20 and 33 are rejected under 35 USC 102(b) as being anticipated by Krauskopf (US Patent No. 5,165,027).

Claims 21-32 are rejected under 35 USC 102(e) as being anticipated by Alverson (US Patent No. 6,480,818).

According to the foregoing, the claims are amended, new claim 34 is added, and, thus, the pending claims remain pending for reconsideration, which is respectfully requested.

No new matter has been added.

**PRIORITY**

In the Office Action page 2, item 3, the Examiner suggests being unable to located the two Japanese foreign priority documents within the electronic file of parent application no. 09/678,732. However, it is noted an image file wrapper (IFW) in the USPTO's PAIR is not available for the parent application, therefore, the foreign priority documents would be in the USPTO's paper file of the parent application, which the Examiner expressly acknowledged receipt of the same in the Notice of Allowance dated July 28, 2003 in the parent application. Accordingly, location and acknowledgment of the two Japanese foreign priority documents is respectfully requested.

**PRIOR ART REJECTIONS**

The independent claims are 14, 15, 21, 27 and 33.

The Examiner maintains from the previous office action the anticipation rejection of independent claims 14, 15 and 33 over Krauskopf.

Regarding Krauskopf, in the Office Action Response to Arguments, the Examiner allegedly interprets a break point instruction to be similar to the claimed "conditional instruction," because a break point instruction is conditional upon whether the stored breakpoint address is enabled for program access. The Examiner relies upon Krauskopf column 4, lines 15-39.

However, a break point instruction is a instruction set to cause a breakpoint, which occurs when the address of the instruction to cause the interrupt is set in a register, and if the set instruction matches an instruction address of an actually to be executed instruction, the interrupt occurs (see, for example, page 7, lines 16-21). In contrast, a conditional statement is a branch instruction that is an actually to be executed instruction fetched from an instruction fetcher for which it is determined whether a designated condition is satisfied, and only when the condition is satisfied, designated processing is executed (see, for example, page 15, lines 12-20). More particularly, an instruction executor (FIGS. 3 and 18-19; 30) executes a conditional or branch instruction, but a set break point instruction is detected rather than executed. See, for example, the present Application page 12, line 7 to page 13, line 6; and page 14, lines 9-20; and page 39, line 20 to page 42, line 5. Accordingly, the Office Action interpretation is not appropriate.

Nevertheless, the independent claims, using claim 1 as an example, are amended to provide:

14. (CURRENTLY AMENDED) An interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated condition of a branch is satisfied and a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

a break detection section for detecting a breakpoint set at an arbitrary position of an instruction sequence;

a condition determination section for determining whether ~~or not~~ a condition of a branch of said conditional instruction is satisfied; and

a control section for controlling a break-interrupt ~~on the basis of~~ based upon a breakpoint detection result from said break detection section and a branch condition determination result from said condition determination section.

The amendments clarify that the claimed "conditional instruction" differs from a breakpoint instruction by expressly providing that the claimed "conditional instruction" is an instruction executing a designated data processing when a designated condition of a branch is satisfied. A breakpoint instruction does not include a determination of a condition of a branch. Accordingly, neither Kauskopf nor Alverson disclose executing the "conditional instruction" and controlling a break-interrupt as claimed.

Krauskopf column 4, lines 15-39 and enable logic circuit and control circuit of FIG. 2, only discuss different breakpoint addresses to be stored representing a reference to data or to the computer program. Contrary to the Office Action Response to Arguments, the claimed embodiment does not only provide 'controlling a break-interrupt on the basis of a breakpoint detection result from said condition determination section,' but the language of the claims expressly provides "a control section for ***controlling a break-interrupt ~~on the basis of~~based upon a breakpoint detection result ... and a branch condition determination result.***" Thus, both ***a breakpoint detection and a branch condition result*** are used to control a break-interrupt.

The Examiner also maintains from the previous office action the anticipation rejection of independent claims 21 and 27 over Alverson. For example, the present application FIG. 18 and the paragraph spanning pages 108-109 support claims 21 and 27. Alverson FIG. 12 and column 3-28 relied upon to reject independent claims 21 and 27, discuss "retrieving ... information on whether or not the breakpoint is conditional. The subroutine continues to step 1210 to determine if the break-point is conditional, and if so continues to step 1215 to evaluate the condition." However, Alverson is directed to a breakpoint handler routine (1125, 1205, 1210) sending a message to a software debugger nub (1215) to evaluate a condition of a conditional breakpoint (1220, 1225). Alverson is directed to a software debugger nub. In contrast, the claimed embodiments support a software debugger that sets an instruction break point for "a conditional instruction that executes a designated data processing when a designated condition of a branch is satisfied and a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible" and/or handles an interrupt during execution of such "conditional instruction that executes a designated data processing when a designated condition of a branch is satisfied" (as the case may be).

A prima facie case of anticipation cannot be established based upon Krauskopf or Alverson by failing to disclose, either expressly or inherently, the claimed "... ***controlling a break-interrupt ~~on the basis of~~based upon a breakpoint detection result ... and a branch condition determination result.***" Thus, both ***a breakpoint detection and a branch condition result*** are used to control a break-interrupt.

NEW CLAIM 34

Further in contrast to Krauskopf and Alverson, the new claimed embodiment provides:

34. (NEW) An apparatus comprising:
- a controller
    - detecting a breakpoint set at an arbitrary position of an instruction sequence;
    - determining a branch of an instruction; and
    - controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction.

Krauskopf and Alverson, respectively, only discuss storing different breakpoint addresses and a breakpoint handler routine (1125, 1205, 1210) sending a message to a software debugger nub (1215) to evaluate a condition of a conditional breakpoint (1220, 1225), and cannot anticipate claim 34 by failing to disclose, either expressly or inherently, "controlling **a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction.**" Thus, new claim 34 is patentably distinguishing over Krauskopf and Alverson.

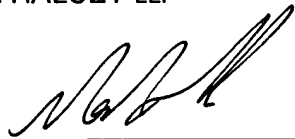
**CONCLUSION**

If there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,  
STAAS & HALSEY LLP

Date: February 28, 2007

By:   
Mehdi D. Sheikerz  
Registration No. 41,307

1201 New York Ave, N.W., 7th Floor  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501